Linear Combinations of Vectors

In the study of signals and signal processing, there is a particular mathematical operation that will show up quite a few times, that of a **linear combination**. Given a collection of vectors in a vector space, say M vectors $x_0, x_1, \dots x_{M-1} \in C^N$, and M scalars $\alpha_0, \alpha_1, \dots, \alpha_{M-1} \in C$, then the linear combination of these vectors is: $y=\alpha_0 x_0 + \alpha_1 x_1 + \dots + \alpha_{M-1} x_{M-1} = \sum_{m=0}^{M-1} \alpha_m x_m$. The result is itself also a vector in the vector space.

Linear Combination Example: A Mixing Board

A linear combination is a scaled sum of different vectors in a vector space. A real world example of a linear combination is the mixing board used in music recording studios. The board takes in a variety of different inputs and combines them--amplifying some, reducing the level of others--to create a single output of music. Mathematically, we could say $x_0=$ drums, $x_1=$ bass, $x_2=$ guitar, \dots, $x_{22}=$ saxophone, $x_{23}=$ singer, and then the mixing board creates the linear combination $y = \alpha_0 x_0 + \alpha_1 x_1 + \dots + \alpha_{M-1} x_{M-1} = \sum_{m=0}^{M-1} \alpha_m x_m$. Changing the different α_m's would result in a different kind of sound that either emphasized or de-emphasizes certain instruments. For example, the producer in the studio that day may be particularly interested in the cowbell.

Linear Combinations as Matrix Multiplication

It is possible to express a linear combination without explicitly using sums, but rather as the product of a matrix and a vector. To do this, all of the vectors to be scaled and summed in the linear combination must first be arranged into a matrix: $\begin{align*} X &= \begin{bmatrix} x_0 | x_1 | \cdots | x_{M-1} \end{bmatrix}\\ &=\begin{bmatrix} x_0[0] &x_1[0] &\cdots &x_{M-1}[0] \\ x_0[1] &x_1[1] &\cdots &x_{M-1}[1] \\ \vdots &\vdots &&\vdots \\ x_0[N-1] &x_1[N-1] &\cdots &x_{M-1}[N-1] \end{bmatrix} \end{align*}$ Then the scaling factors for each vector are stacked into a single vector: $a = \begin{bmatrix}\alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{M-1} \end{bmatrix}$ Once those are in place, the linear combination is ultimately expressed as a simple matrix-vector

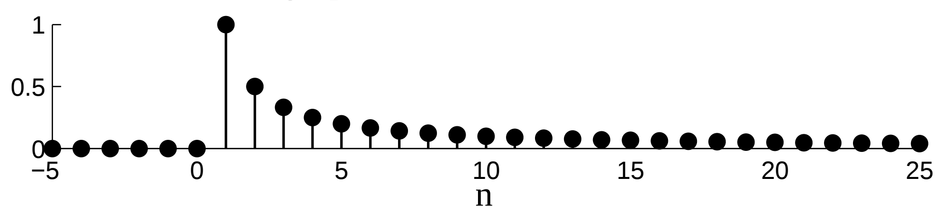
multiplication: $y=\alpha_0 x_0 + \alpha_1 x_1 + \dots + \alpha_{M-1} x_{M-1} = \sum_{m=0}^{M-1} \alpha_m x_m = \begin{bmatrix}x_0 | x_1 | \cdots | x_{M-1} \end{bmatrix}\begin{bmatrix}\alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{M-1} \end{bmatrix}= X a$

Norms and Inner Products of Infinite Length Vectors

Like as with finite length signals (which can be understood as vectors in R^N or C^N), we can take the norms and inner products of infinite length signals, as well. For the most part, the concepts will apply just the same.

The Norms of Infinite Length Vectors
As with finite-length vectors, the p-norm for infinite-length vectors is a sum: $\|x\|_p = \left(\sum_{n=-\infty}^{\infty} |x[n]|^p \right)^{1/p}$ Unlike with the p-norms of finite-length signals, the summation for infinite length vectors is necessarily fininite. As a consequence, the p-norms of infinite length signals may not be bounded. Consider a finite-length signal $x[n]=1$ for $0\leq n\lt N-1$. The 2 norm for this signal would be $\sqrt{N}$. However, for an infinite length signal of constant value 1, all the p norms will be unbounded!

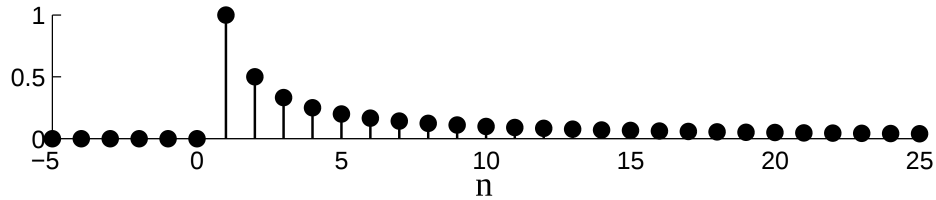
For some infinite length vectors, the p norm may only exist for certain values of p. Consider the signal: $x[n] = \begin{cases} 0 & n\leq0 \\ \frac{1}{n} & n\geq 1 \\ \end{cases}$



The 2-norm of this signal exists: $\|x\|_2^2 = \sum_{n=-\infty}^{\infty} |x[n]|^2 = \sum_{n=1}^{\infty} \left|\frac{1}{n}\right|^2 = \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \approx 1.64$ But the 1-norm is unbounded: $\|x\|_1 = \sum_{n=-\infty}^{\infty} |x[n]| = \sum_{n=1}^{\infty} \frac{1}{n} = \infty$

For a finite-length vector, the ∞ norm is simply the maximum value of the magnitudes of all its elements. For infinite-length vectors, it is instead the **supremum**, or the **least upper bound** of the set of all the elements of the vector. The distinction between that value, and the maximum value, of the set is outside the bounds of this course. For all the cases we will consider, the ∞ norm will simply be the largest of the

magnitudes of the elements of the vector. For example, for the signal $x[n]$ below, $\|x\|_\infty=1$:



Inner Product of Infinite Length Signals

The inner product of infinite length signals is defined the same way as for finite-length signals, except that the sum is infinite: $\langle x, y \rangle = \sum_{n=-\infty}^{\infty} x[n] \, y[n]^*$ The angle is also the same: $\cos \theta_{x,y} = \frac{\Re{\langle x, y \rangle}}{\|x\|_2 \, \|y\|_2}$

Linear Combinations of Infinite Length Vectors

When it comes to infinite-length vectors, we will also be interested in linear combinations of them. However, in contrast with finite-length vectors, we will be interested in the linear combination of an infinite number of the vectors: $y = \sum_{m=-\infty}^{\infty} \alpha_m x_m$ As with finite-length vectors, infinite-length vectors can be stacked into a matrix (an infinitely large one) and be multiplied by an infinite length vector of scalers to perform linear combination: $X = \begin{bmatrix} \cdots | x_{-1} | x_0 | x_1 | \cdots \end{bmatrix}$ $a = \begin{bmatrix} \vdots \\ \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \\ \vdots \end{bmatrix}$ $\begin{align*}y&= \sum_{m=-\infty}^{\infty} \alpha_m x_m &= \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & x_{-1}[-1] & x_{0}[-1] & x_{1}[-1] & \cdots \\ \cdots & x_{-1}[0] & x_{0}[0] & x_{1}[0] & \cdots \\ \cdots & x_{-1}[1] & x_{0}[1] & x_{1}[1] & \cdots \\ & \vdots & \vdots & \vdots & \\ \end{bmatrix} \begin{bmatrix} \vdots \\ \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \\ \vdots \end{bmatrix} &= X a$

Inner Products and Orthogonality

In discrete-time signal processing, understanding signals as vectors within a vector space allows us to use tools of analysis and linear algebra to examine signal properties. One of the properties we may want to consider is the similarity of (or difference between) two vectors. A mathematical tool that provides insight into this is the **inner product**.

Transposing Vectors

One of the ways to express the inner product of two vectors is through matrix multiplication, so first we must introduce the concept of transposing vectors. In order to multiply two matrices (a vector is simply a matrix in which one of the dimensions is 1), the column dimension of the first must match the row dimension of the second. To make those match for two vectors of the same length, we must **transpose** one of them. To take the transpose of a matrix, simply turn the rows into columns: the first row will become the first column in the transposed matrix; the second row, the second column, and so on. Here is how that looks for a vector. An N-row, single column vector transposed becomes a 1 row, N-column vector: $\begin{bmatrix}x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}^T = \begin{bmatrix}x[0] & x[1] & \cdots & x[N-1] \end{bmatrix}$

Now, when it comes to complex valued vectors, we can take a transpose in the same way, but for the purposes of finding an inner product we actually need to take the conjugate, or Hermitian, transpose, which involves taking the transpose and then the complex conjugate: $\begin{bmatrix}x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}^H =\begin{bmatrix}x[0]^* & x[1]^* & \cdots & x[N-1]^* \end{bmatrix}$ Of course, for real valued vectors, the regular transpose and Hermitian transpose are identical.

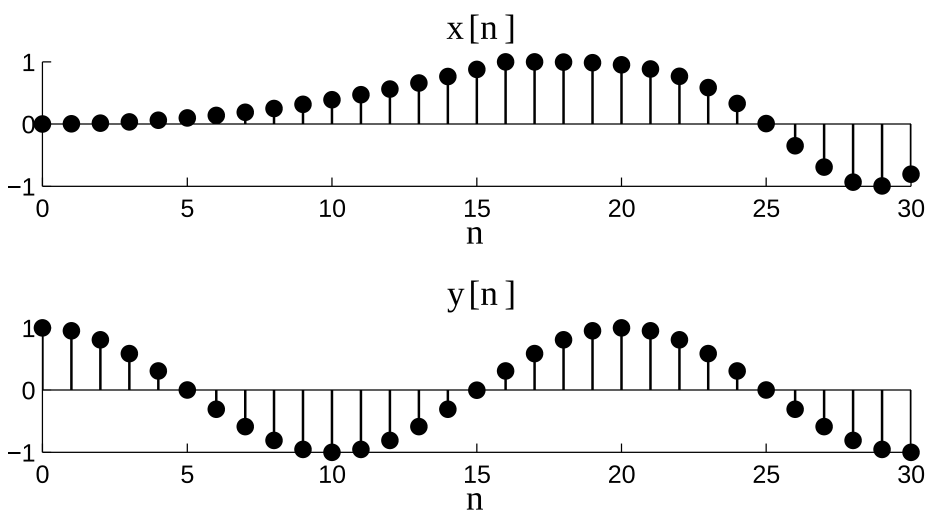
The Inner Product

The inner product of two complex (or real) valued vectors is defined as: $\langle x, y \rangle = y^H x = \sum_{n=0}^{N-1} x[n] \, y[n]^*$ So the inner product operation takes two vectors as inputs and produces a single number. It turns out that the number it produces is related to the angle θ between the two vectors: $\cos \theta_{x,y} = \frac{\Re{\langle x, y \rangle}}{\|x\|_2 \, \|y\|_2}$ This formula works for complex and real

vectors, although taking the real part of the inner product is redundant for real valued ones.

For two (or three) dimensional vectors, this angle is exactly what you would expect it to be. Let $x=\begin{bmatrix}1 \\ 2 \end{bmatrix}$, ~ $y=\begin{bmatrix}3 \\ 2 \end{bmatrix}$ We have $\|x\|_2^2 = 1^2 + 2^2 = 5$, $\|y\|_2^2 = 3^2 + 2^2 = 13$,and $\langle x, y \rangle=(1)(3)+(2)(2)=7$. The angle between them is $\arccos\left(\frac{7}{\sqrt{(5)(13)}}\right) \approx 0.519~{\rm rad} \approx 29.7^\circ$. If you plot the vectors out in the Cartesian plane, you will indeed see an angle between them of about 30 degrees.

For higher dimensional signals the result of the inner product--how it relates to the angle between signals--may not seem as intuitive, but the information it provides is still just as useful, and of course it is computed in the same was as with shorter vectors. Consider the signals below:

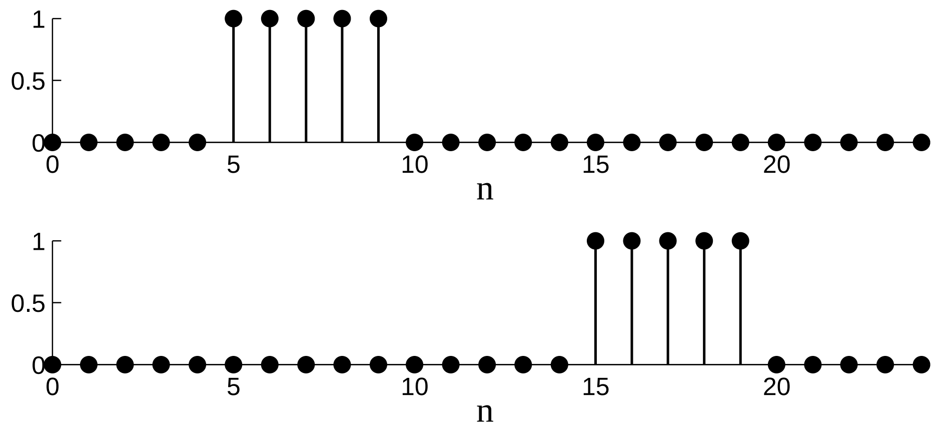


The inner product of these two signals, computed according to the formula above, is $\langle x, y \rangle ~=~ y^T x ~=~ 5.995$, which corresponds to an angle of $\theta_{x,y} ~=~ 64.9^\circ$.

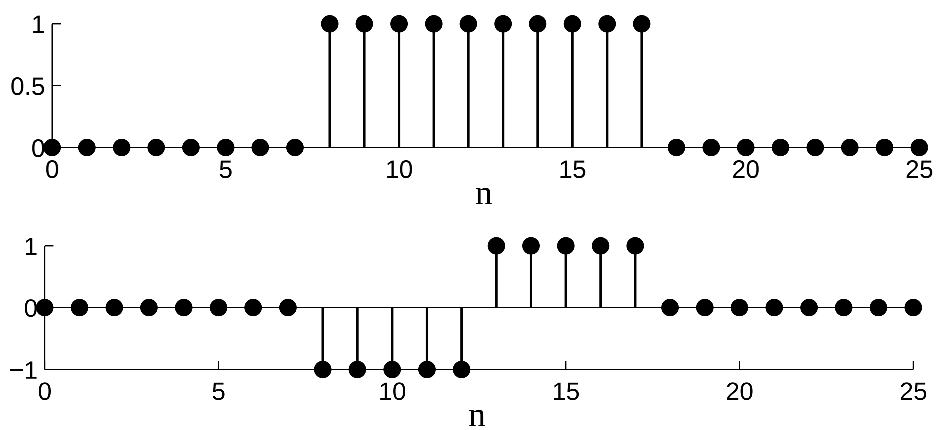
Inner product: Limiting Cases
Recall that the inner product is defined as a sum ($\sum_{n=0}^{N-1} x[n] \, y[n]^*$), which can also be expressed as a vector product ($y^H x$). Let's look at a couple of interesting values that sum could take.

The dot product of two signals could be rather large. If the signals are identical, it is simply the norm of the signal, squared: $\langle x, x \rangle = \sum_{n=0}^{N-1} x[n] \, x[n]^* = \sum_{n=0}^{N-1} |x[n]|^2 = \|x\|_2^2$.

On the other hand, it is also possible for the dot product sum to be 0. Consider the two signals below:



The inner product of those two signals is obviously zero because each pointwise product is also zero. But it is possible, of course, for products in the sum to be nonzero and still have the total add up to zero:



Whenever the inner product of two signals is zero, it is defined that those signals are **orthogonal**.

Orthogonality of Harmonic Sinusoids

Recall the special class of discrete-time finite length signals called harmonic sinusoids: $s_k[n] = e^{j \frac{2 \pi k}{N} n}, ~~~

n,k,N\in\Integers, ~~ 0\leq n \leq N-1, ~~ 0\leq k \leq N-1$ It is a very interesting property that any two of these sinusoids having different frequencies (i.e., $k\neq l$) are orthogonal: $\begin{align*} \langle s_k | s_l \rangle &= \sum_{n=0}^{N-1} d_k[n] d_l^*[n]\\ &= \sum_{n=0}^{N-1} e^{j \frac{2 \pi k}{N} n} (e^{j \frac{2 \pi l}{N} n})^*\\ &= \sum_{n=0}^{N-1} e^{j \frac{2 \pi k}{N} n} \: e^{-j \frac{2 \pi l}{N} n} \\ &= \sum_{n=0}^{N-1} e^{j \frac{2 \pi}{N} (k-l) n} ~~~ \textrm{let }r=k-l \in I, r\neq 0 \\ &= \sum_{n=0}^{N-1} e^{j \frac{2 \pi}{N} r n}\\ &=\sum_{n=0}^{N-1} a^n ~~~ \textrm{with }~a=e^{j \frac{2 \pi}{N} r},\textrm{and recall }\sum_{n=0}^{N-1} a^n = \frac{1-a^N}{1-a} \\ &= \frac{ 1- e^{j \frac{2 \pi r N }{N}} }{1-e^{j \frac{2 \pi r}{N}} } ~=~ 0 ~~\checkmark \end{align*}$ If two of these sinusoids have the same frequency, then their inner product is simply N: $\begin{align*} \|s_k \|_2^2 &= \sum_{n=0}^{N-1} |d_k[n]|^2 \\ &= \sum_{n=0}^{N-1} |e^{j \frac{2 \pi k}{N} n}|^2\\ &=\sum_{n=0}^{N-1} 1 ~=~ N\checkmark \end{align*}$

So the dot product of two harmonic sinusoids is zero if their frequencies are different, and N if they are the same. In order to make latter number is 1, instead of N, they are sometimes normalized: $\tilde{d}_k[n] = \frac{1}{\sqrt{N}}\, e^{j \frac{2 \pi k}{N} n}, ~~~ n,k,N\in Z, ~~ 0\leq n \leq N-1, ~~ 0\leq k \leq N-1$

Matrix Multiplication and Inner Products
Let's take look at the formula for the matrix multiplication $y=Xa$. For notation, we will represent the value on the nth row and mth column of X as $x_m[n]$. The matrix multiplication looks like this: \begin{bmatrix}\vdots & \vdots && \vdots \\ x_0[n] & x_1[n] & \cdots & x_{M-1}[n] \\ \vdots & \vdots && \vdots \end{mbatrix} And the value for the multiplication is: $y[n] = \sum_{m=0}^{M-1} \alpha_m \, x_m[n]$ Hopefully that formula looks familiar! What it is showing is that the matrix multiplication $y=Xa$ is simply the inner product of a with each row of X, when X and a are real. If they are complex valued, then the complex conjugate of one of them would have to be performed before the matrix multiplication for each value of y to be the inner product of the matrix row with a.

The Cauchy Schwarz Inequality

Above we saw that the inner product of two vectors can be as small as 0, in which case the vectors are orthogonal, or it can be large, such as when the two vectors are identical (and the inner product is simply the norm of the vector, squared). It turns out that there is a very significant inequality that explains these two cases. It is called the **Cauchy Schwarz Inequality**, which states that for two vectors x and y, $|\langle x,y \rangle |\leq |\x\| \|y\|$ Now the magnitude of the inner product is always greater than or equal to 0 (being 0 if the vectors are orthogonal), so we can expand the inequality thus: $0\leq |\langle x,y \rangle |\leq |\x\| \|y\|$ If we divide the equation by $|\x\| \|y\|$, then we have $0\leq \frac{|\langle x,y \rangle |}{|\x\| \|y\|}\leq 1$ This explains why we can define $\cos \theta_{x,y} = \frac{\Re{\langle x, y \rangle}}{\|x\|_2 \, \|y\|_2}$, for the cosine function also has a range of 0 to 1.

Now there are many different proofs of the inequality, and it is something of a mathematical pastime to appreciate their various constructions. But for signal processing purposes, we are more interested in the utility of the inequality. What it basically says is that--when the lengths of two vectors are taken into consideration--their inner product ranges in value from 0 to 1. Because of this, we can see that the inner product introduces some kind of comparison between two different vectors. It is at its smallest when they are, in a sense, very different from each other, or **orthogonal**. It is at its peak when the vectors are simply scalar multiples of each other, or in other words, are very alike.

It turns out there are many application in which we would like to determine how similar one signal is to another. How does a digital communication system decide whether the signal corresponding to a "0" was transmitted or the signal corresponding to a "1"? How does a radar or sonar system find targets in the signal it receives after transmitting a pulse? How does many computer vision systems find faces in images? For each of these questions, the similarity/dissimilarity bounds established by the Cauchy Schwarz inequality help us to determine the answer.

Signals are Vectors

One mathematical way of understanding signals is to see them as functions. A signal $x[n]$ carries some kind of information, having a value $x[n]$ at every given point of its independent time variable n. Another, and complementary, way of understanding signals is to consider them as vectors within vector spaces. By doing this we will be able to apply various tools of linear algebra to help us better understand signals and the systems that modify them.

Vector Spaces

A **vector space** V is a collection of vectors such that if $x, y \in V$ and α is a scalar then $ \alpha x \in V \text{and} x+y \in V $ In words, this means that if two vectors are elements of a vector space, any combination or scaled version of them is also in the space. When we consider the scaling factors, we mean α that are either real or complex numbers.

There are many different kinds of vector spaces, but the two in which we are especially interested are R^N and C^N. R^N is the set of all vectors of dimension N, in which every entry of the vector is a real number, and C^N is exactly the same, except the entries are complex valued.

Starting Small – A Two Dimensional Vector Space

You are already familiar with a prominent example of a vector space, the two-dimensional real coordinate space R^2. Every ordered combination of two real numbers is a vector in this space, and can be visualized as a point or arrow in the two-dimensional Cartesian plane. Suppose x and y are each vectors in R^2. $x=\begin{bmatrix} x[0] \\ x[1] \end{bmatrix} $y=\begin{bmatrix} y[0] \\ y[1] \end{bmatrix} where $x[0], x[1], y[0], y[1]\in R$. The indices we use to refer to elements of the vector start their numbering at 0. This is the common convention in signal processing and many mathematical languages, like C, but note that vector indices in MATLAB start with 1. Scaled versions of these vectors are still within the space: $\alpha x=\alpha \begin{bmatrix} x[0] \\ x[1] \end{bmatrix}=\begin{bmatrix} \alpha x[0] \\ \alpha x[1] \end{bmatrix}\in R^2$ So is the sum of two vectors: $x+y=\begin{bmatrix} x[0] \\ x[1]

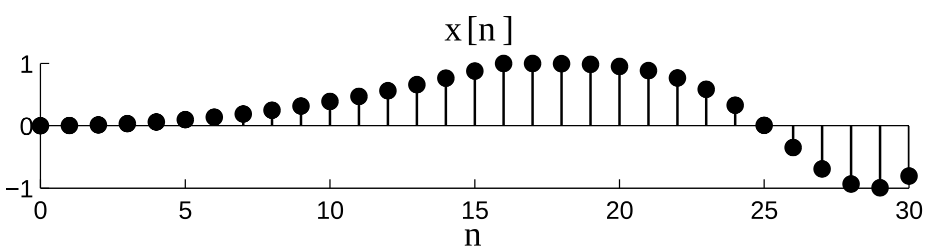
\end{bmatrix}+\begin{bmatrix} y[0] \\ y[1] \end{bmatrix}=\begin{bmatrix} x[0]+y[0] \\ x[1]+y[1] \end{bmatrix}\in R^2$

The Vector Space R^N

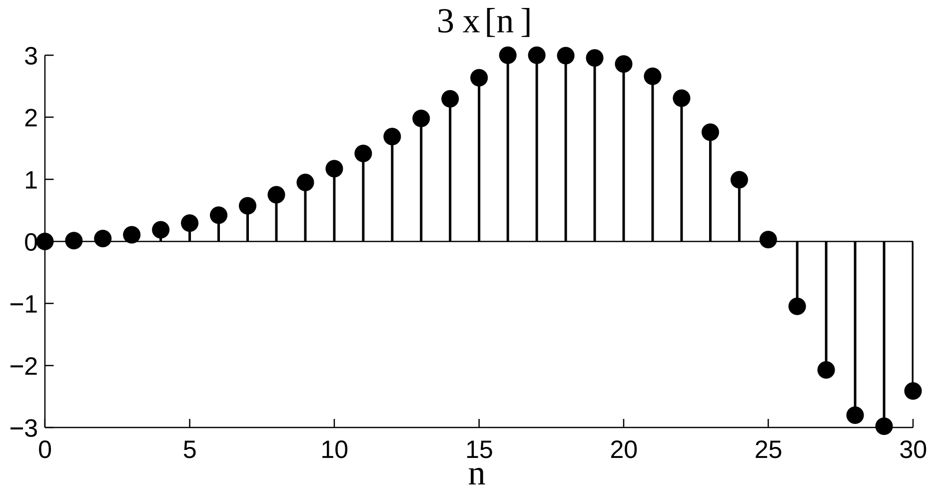
Let's now move from two dimensional vectors to those with N dimensions, each taking a real valued number: $x=\begin{bmatrix}x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}$, where $x[n]\in\Real$ So we have a mathematical entity x, with N ordered real values associated with it. Stated this way, we see that x is a signal, simply expressed in a vector form x as opposed to the signal notation form $x[n]$, but both forms refer to the exact same thing.

Just as with in 2-dimensions, we can perform the same operations on the N-dimensional signal/vector x:

When a vector x in R^N is scaled by $\alpha\in R$, the result is still in R^N.



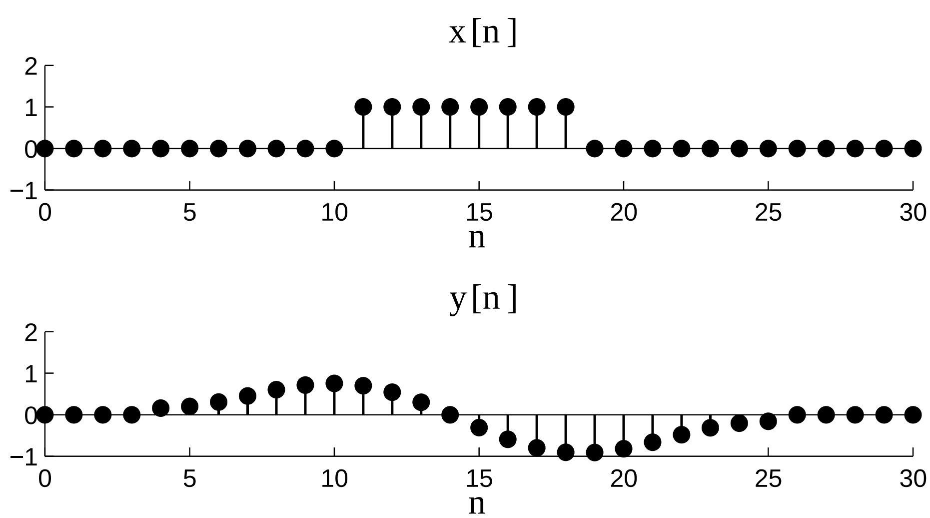
An example of an N-dimensional signal/vector x.

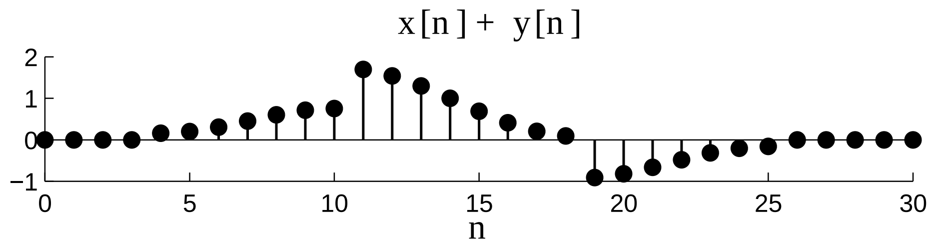


The signal x, scaled by $\alpha=3$.

When each of the real-valued elements of x is scaled by a real value α, the results are still real-valued, of course, so the resulting scaled vector is still within R^N. As R^N is a vector space, that should come as no surprise, and neither should the fact that the sum of two vectors in R^N is itself another vector in the vector space R^N:

The sum of two vectors in R^N is another vector in R^N.





The Vector Space C^N

Ordered sequences of N complex numbers form the vector space C^N, just like their real counterparts do in R^N: $x=\begin{bmatrix}x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}$, where $x[n]\in C$ So each element of a vector in C^N is a complex number, which can be represented either in terms of its real and imaginary parts, or in terms of its magnitude and phase. The vector as a whole can also be represented in those ways, either in rectangular form Rectangular form $x = \begin{bmatrix} \Re{x[0]} + j \Im{x[0]} \\ \Re{x[1]} + j \Im{x[1]} \\ \vdots \\ \Re{x[N-1]} + j \Im{x[N-1]} \end{bmatrix} = {\rm Re} \left\{ \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}\right\} + j\: {\rm Im} \left\{ \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}\right\} $ or in polar form $ x = \begin{bmatrix} |x[0]|\,e^{j \angle x[0]} \\ |x[1]|\,e^{j \angle x[1]} \\ \vdots \\ |x[N-1]|\,e^{j \angle x[N-1]} \end{bmatrix} $.